Appendix B – DynaMO Results

The following graphs illustrate the benefits of DynaMO compared to Pastry in a little more detail. These graphs only have very brief descriptions. For a detailed description and even more results please refer to [5].

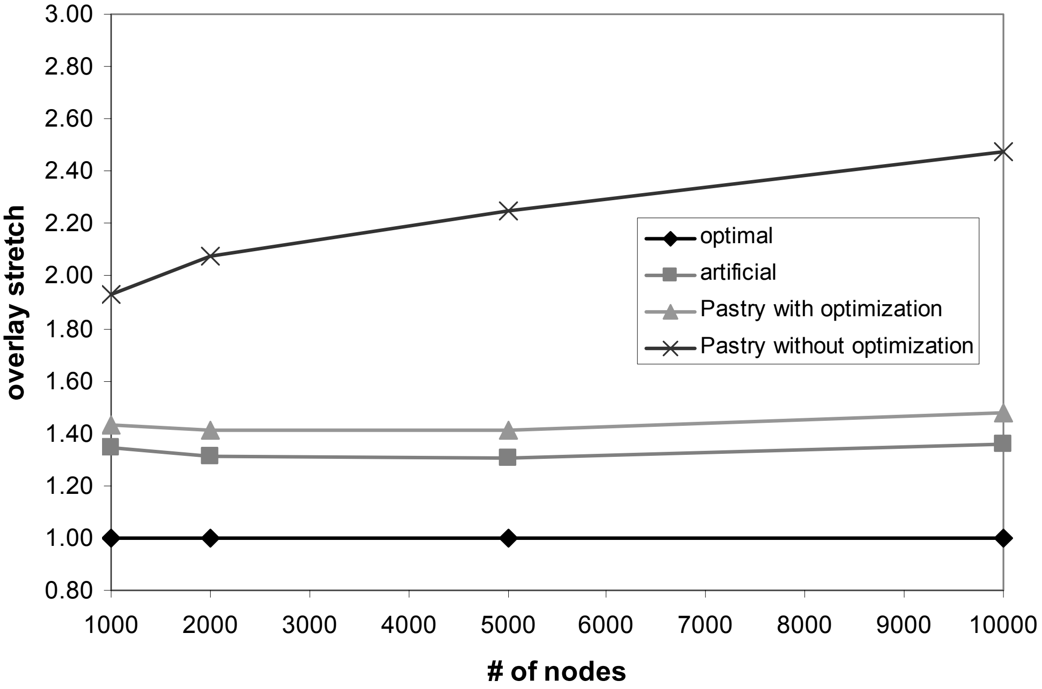

Fig. B.1 Overlay stretch of the various Pastry bootstrap mechanisms

The figure above shows the various overlay stretches Pastry achieves with different bootstrap mechanisms. Optimal is of course if the overlay lookup process is as long as the direct physical path. "Artificial" denotes a bootstrap algorithm in which every node's overlay routing table is filled employing exact global knowledge and therefore represents the theoretical best case scenario. The other two graphs show the overlay stretch when using Pastry's inbuilt optimization technique and when not using it respectively.

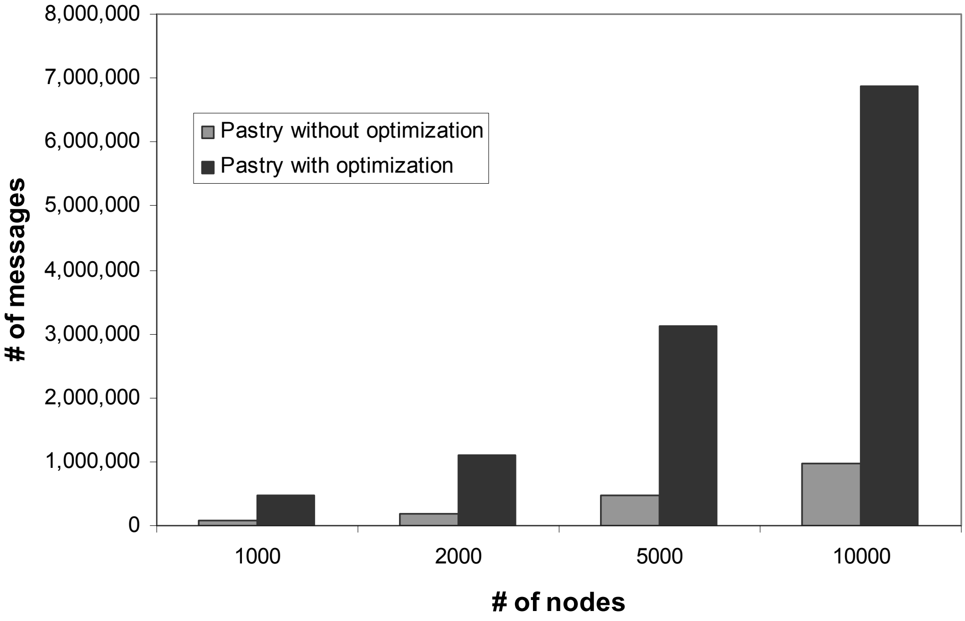

Fig. B.2 Total number of messages exchanged during bootstrap (Pastry)

Fig. B.2 illustrates why there is such a big difference between the overlay stretches of the optimized bootstrap version and the un-optimized Pastry overlay. The optimization algorithm creates a huge amount of overhead to optimize routing table entries.

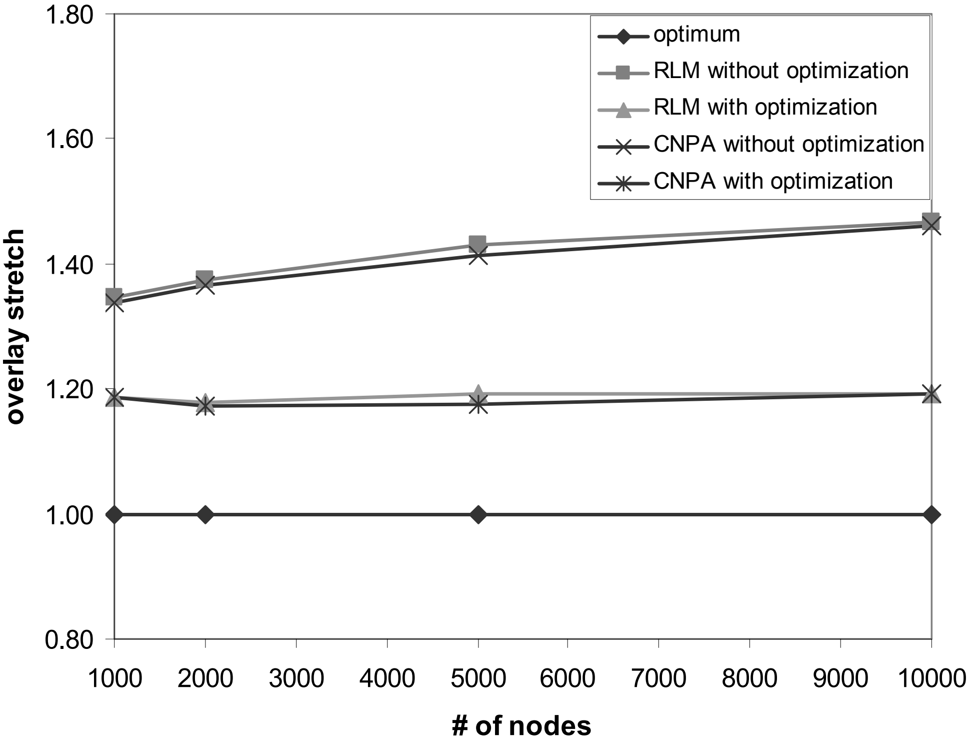

Fig. B.3 Overlay stretch achieved by DynaMO

DynaMO achieves a significantly lower overlay stretch even without optimization as Fig. B.3 shows. As DynaMO is based on Pastry it can employ the same optimization technique and is able to lower the overlay stretch respectively under Pastry's theoretical optimum.

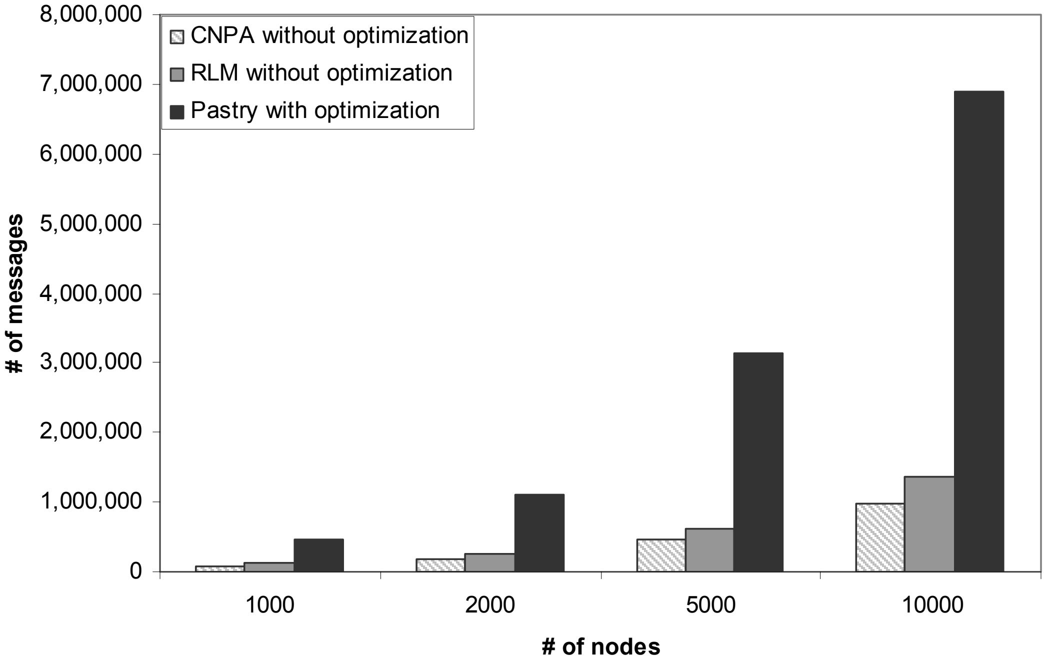

Fig. B.4 Total number of messages exchanged during bootstrap (Pastry, RLM, CNPA)

The figure above illustrates the bootstrap overhead of three different overlay networks with comparable overlay stretches. As clearly can be seen, both approaches implemented by DynaMO outperform Pastry significantly.